

### **III. REMARKS**

#### **Status of the Claims**

Claims 1, 15, 16, 29, 30, 32-38, and 40-43 are amended. Claim 63 is cancelled. Claims 1, 2, 4-10, 12-16, 18-24, 26-30, 32-38, 40-43, 45-58, 61, and 62 are presented for further consideration.

#### **The Office Action and Responsive Remarks**

The claims of this application are amended as indicated above. These amendments are for the purpose of clarification. No new matter is presented. The amendments to the claims are not intended to be limiting, are not made for reasons related to patentability, and should not be interpreted to raise issues of estoppel.

Applicant has considered the Examiner's comments set forth in the Office Action mailed February 4, 2009 and responds in detail below. Reconsideration of the application is respectfully requested in view of the amendments and the following remarks.

#### **The Office Action**

Claims 15-16, 18-24, 26-30, 32-38, 40-42, and 63 stand rejected under 35USC101. Applicant submits that the amendments to the claims fully remedy the basis for this rejection.

Claims 1,9,12,14,15,23,26,28,29,37,40,42,43,46,54,56, and 61-63 are rejected under 35USC103(a) based on the combined teaching of the reference Meppelink et al, U.S. Patent No. 5,542,063 and Sullivan, U.S. Patent No. 5,737,557 and further in view of the cited reference Balsara, U.S. Patent No. 6,065,012. This rejection is traversed on the following grounds:

The combined teaching of Meppelink, Sullivan, and Balsara does not render claims 1,9,12,14,15,23,26,28,29,37,40,42,43,46,54,56, and 61-63 obvious because it fails to teach or otherwise suggest each and every limitation of the independent claims of this

application. It is well settled that in order to establish a prima facie case for obviousness, the prior art reference (or references when combined) must teach or suggest all the claim limitations. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, without reference to the disclosure of this application. (MPEP Section 2142) ***In re Vaeck*, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991). See MPEP § 2143 - § 2143.03 for decisions pertinent to each of these criteria."**

Meppelink discloses an input device emulator that comprises an interpreter for a windowed environment. The interpreter reads and executes commands from a file. The interpreter commands correspond to a variety of user interface control device actions and relate to the manipulation of the windowed environment and the entry of text via a keyboard. The interpreter simulates an actual user for mass-testing purposes.

Meppelink also discloses that the windowed environment comprises an application program that communicates with a window server program. The window server program constructs window user interface elements and represents information obtained from the application program in these window user interface elements. The window server controls the display using lower level user interface display instructions. Normally, input from a user via an I/O device is processed in the window server program to map the user input, for example, mouse clicking at specified locations on the display to user interface events raised to the application program such as the selection of a menu item. However, in order to support mass-testing of user interfaces, an input device emulator is coupled to the window server to simulate user actions. The user actions are read from a test script. The benefit of placing a user interface emulator between the stream modules and the window server is that the fact, whether a user interface action indication originates from an actual user or from an emulator, script may be hidden from the window server. Thus, software testing may be made easier.

Meppelink fails to disclose the use of the method in production mode, that is, in normal usage.

Sullivan discloses a windowed environment where a number of collective actions may be associated with the files contained in a software suite. A software suite is a file and program folder displayed to the user as an icon that may be expanded to a window by user selection (Sullivan, column 5, row 58—60). It comprises files from different file system directories. Each software suite has associated with it a storage element in which is stored contextual information for defining the appearance and behavior of the software suite window. A spot interface is designed to float on top of the desktop and all windows that are currently open. The placement of the mouse pointer symbol over the spot interface results in a display of a circular visual element that encircles the spot interface. The circular visual element, in other words, a collar, comprises four quadrants each of which correspond to a secondary user interface. The selection of a quadrant in the collar results in the display of a secondary user interface that comprises a number of icons on the secondary user interface that are used to start an application or to open a file when the icon is clicked. To summarize, Sullivan discloses the mere fact that a program or a file may be started or opened by clicking a mouse button while the mouse cursor is over the corresponding icon.

The Examiner acknowledges that with respect to independent claims 1,15,29, and 43, combined teaching of Meppelink and Sullivan fails to the following features:

“determining a view chain data structure comprising at least three entries, each of said entries comprising an application identifier and a view identifier, a view identified by said view identifier being within an application identified by said application identifier;.....

checking whether entries for views not launched remain in said view chain data structure, each said entry for a view not launched specifying a view identifier for a view not yet launched by said view router;

launching a second view based on a second entry in said view chain data structure automatically by said view router when entries for views not launched remain in said view chain; and

continuing said first application when no entries for views not launched remain in said view chain data structure.”

In order to overcome this acknowledged deficiency, the Examiner relies on the reference Balsara.

Balsara discloses a method for managing user-relevant data such as E-mail, calendar, contact and task list data. The user-relevant data is gathered from a multitude of applications. The method comprises the forming of a summary HTML page that consists of multiple separate tables. There is one table for each application. Each application is adapted to present its data in a table on the summary HTML page, thereby providing a specific method for gathering the data to be presented. The summary HTML page source code comprises tags for defining each of the tables. In each table tag, there is a reference to an object class that acts as a data source for the data to be fetched to the table rows and columns. The object classes are defined by the parameter names corresponding to the table columns. The browser invokes a method in the data source object class upon encountering a table definition in the HTML source code. An entire application is not invoked when a table presenting its data is filled, instead, only the data source method defined in the application is invoked. The tables are also dynamic in the sense that they are updated to display changed data by monitoring for trigger events indicating that the underlying data has changed.

Regarding amended claims 1, 36, 42, 46 and 60 the applicant respectfully submits that Meppelink, Sullivan and Balsara fail disclose or suggest the combination of features of determining a view chain data structure comprising at least three entries, each of said entries comprising an application identifier and a view identifier, a view identified by said view identifier being within an application identified by said application identifier;

passing said view chain data structure to a view router from a first application; launching a first view based on a first entry in said view chain data structure automatically by said view router; checking whether entries for views not launched remain in said view chain data structure, each said entry for a view not launched specifying a view identifier for a view not yet launched by said view router; launching a second view based on a second entry in said view chain data structure automatically by said view router when entries for views not launched remain in said view chain; and continuing said first application when no entries for views not launched remain in said view chain data structure.

As the Examiner admits in the Office Action of 4 February 2009, Meppelink and Sullivan are silent on "determining a view chain data structure comprising at least three entries, each said entries comprising an application identifier and a view identifier; a view identified by said view identifier is associated with an application identified by said application identifier; checking whether unprocessed entries remain in said view chain data structure; launching a second view based on a second entry in said view chain data structure automatically by said view router when unprocessed entries remain in said view chain; and continuing said first application when no unprocessed entries remain in said view chain data structure".

The applicant respectfully submits that Balsara fails to disclose the above features of the independent claims.

In Balsara a view is an entire HTML page (Balsara, Figure 3, reference numeral 210) which comprises individual HTML tables therein (Balsara, Figure 3, reference numerals 215a — 215d). The tables are displayed simultaneously. The tables are filled using application specific data source methods (Balsara, column 14, rows 11 — 64) that provide mere data values. In Balsara the data source methods do not represent views, that is, user interfaces provided ready from within an application. Balsara fails to

disclose the launching of views that are identified views within an application. In Balsara there is only a single entity that is called a view, namely the entire HTML page within a browser. Balsara fails to disclose application specific, separately launched views.

Furthermore, Balsara fails to disclose the features indicated above relating to the checking whether entries for views not launched remain in said view chain data structure, each said entry for a view not launched specifying a view identifier for a view not yet launched by said view router; launching a second view based on a second entry in said view chain data structure automatically by said view router when entries for views not launched remain in said view chain; and continuing said first application when no entries for views not launched remain in said view chain data structure.

In Balsara it is merely checked if data associated with a given table on the HTML page has changed. If the data has changed, the table is refreshed with the modified data. Balsara does not disclose that a view chain data structure is checked for entries for views not launched, the entries specifying a view identifier for a view not yet launched. As stated hereinbefore, the interpretation of a view in Balsara is different. In Balsara the single view is displayed as the HTML page is rendered on the screen (Balsara, Figure 2, reference numeral 210). In Balsara there is no view chain data structure that identifies views that have not yet been launched, the views being within applications. It should also be noticed that the object classes for obtaining selected data from an application for filling an HTML table on the HTML page do not comprise the launching of a view from that application. The object classes in Balsara are for obtaining named parameter values from an application, not for launching views or windows provided within that application (Balsara, column 14, rows 11 — 64). Further, Balsara does not use a view chain data structure for determining whether there are views not yet launched.

Finally, Balsara does not disclose that a first application is continued when no entries for views not launched remain in said view chain data structure. Even if the browser is interpreted as the first application, in Balsara the displaying of the HTML page is continued in the browser even as data source methods are invoked on applications. The HTML tables are filled by the browser with the obtained data directly after the invocation of the method. Thus, there is no step, in the method disclosed by Balsara, that teaches the step of continuing said first application when no entries for views not launched remain in said view chain data structure. The reference Balsara, therefore, fails to remedy the acknowledged deficiencies of the combined teaching of Meppelink and Sullivan.

For all the foregoing reasons, it is respectfully submitted that the amended independent claims 1, 15, 29 and 43 are patentable over Meppelink, Sullivan and Balsara.

Since each of the independent claims of the present application are believed to be distinguished over the cited art, it is respectfully submitted that pending dependent claims which depend from independent claims 1, 15, 29 and 43 are further distinguished over the cited art and contain patentable subject matter.

In paragraph 9 of the office action, claims 2, 4-8, 10, 13, 16, 18-22, 24, 27, 30, 32-36, 38, 41, 45, 47, 49-53, 55, and 57 are rejected under 35USC103(a) based on the combined teaching of Meppelink, Sullivan, and Balsara and further in view of Bahrs, U.S. Patent No. 7,181,686. This rejection is traversed on the following grounds:

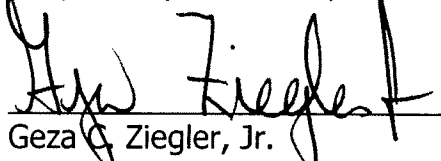
The combined teaching of Meppelink, Sullivan, and Balsara in view of Bahrs does not render claims 2, 4-8, 10, 13, 16, 18-22, 24, 27, 30, 32-36, 38, 41, 45, 47, 49-53, 55, and 57 obvious because it fails to teach or otherwise suggest each and every limitation of the claims. In particular the combined teaching fails to disclose or suggest the claimed features of independent claims 1, 15, 29, and 43 as indicated above. These grounds apply equally to the rejected dependent claims, all of which, by dependency,

have the limitations described in the independent claims. The cited reference Bahrs fails remedy the deficiencies of the primary combined teaching of Meppelink, Sullivan, and Balsara.

For all of the above reasons, it is respectfully submitted that all of the claims now present in the application are clearly novel and patentable over the prior art of record, and are in proper form for allowance. Accordingly, favorable reconsideration and allowance is respectfully requested. Should any unresolved issues remain, the Examiner is invited to call Applicants' attorney at the telephone number indicated below.

The Commissioner is hereby authorized to charge payment for any fees associated with this communication or credit any over payment to Deposit Account No. 16-1350.

Respectfully submitted,

  
Geza C. Ziegler, Jr.  
Reg. No. 44,004

30 April 2009  
Date

Perman & Green, LLP  
425 Post Road  
Fairfield, CT 06824  
(203) 259-1800  
Customer No.: 2512